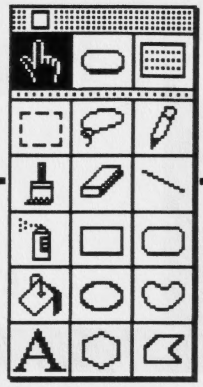
The Apple HyperCard User Group

Issue #6 • March 4, 1988

Windoid



A Publication for the Informed HyperCard User

EDITORIAL

by David Leffler

Well, finally! Just when people were starting to doubt whether or not WINDOID #6 would ever appear, here it is. We have been very busy getting HyperCard 1.1 out the door, and we wanted to make sure that all the Apple dealers and licensed Apple User Groups had the HyperCard 1.1 Upgrade disk before publication. By now, all Apple dealers definitely have the HyperCard 1.1 Update disk in their stores but they are under no obligation to give it to anyone. However, Apple dealers are the best and I am sure they will do everything possible to make sure their customers are satisfied. HyperCard 1.1 contains many user-requested fixes and enhancements to make your stack use, and creation, just that much more elegant. It contains a READ ME FIRST! stack that outlines the new features and updates your Home card with the new Text Arrows check box and some new Import and Export buttons you have asked for. Additionally, there are new and improved internationalized Phone, Area Codes, and Datebook stacks. We were going to ship a new Home stack but recognized that many people had modified their Home cards. If we shipped a new Home stack, some people would have dragged the new Home into their HyperCard Stacks folder and inadvertently replaced their old Home. We hope you like the way we make the transition easy for you. The Text Arrows check box allows you to choose whether or not you want to use cursor keys in text fields. With the box checked you can use your cursor keys to move about in text fields. If you then want to go from card to card just press the Option key first. Naturally, if the box is not checked, everything works just the way you are used to. For all of you that sent in requests for features and found for us some of the problems we fixed for this release, we offer our thanks. You really made a quality difference! If you want the new release, and believe me you do, bring to your Apple dealer either your original HyperCard Startup, HyperCard & Stacks disks, or the purchase slip for your Macintosh dated after August 11, 1987, and the Upgrade is FREE. Also, if you want to keep your dealer happy, please bring in an INITIALIZED disk for the upgrade. They are not going to give disks away. NOTE: When we enhanced the 1.1 Datebook stack to work with the international HyperCard version, there was a modification that makes a small part of the HyperCard manual tutorial not work properly. When you try to create the To Do stack from the Datebook stack, the entire Datebook stack gets created. This can be a bit disconcerting. The best thing to do is either copy the To Do card and paste it into a new empty stack, or just use the old Datebook stack to do the tutorial.

In This Issue

- Editorial: HyperCard 1.1
- Editors Choice: MacRecorder™
- Power Tips Six by Phil Wyman
- Gesture by Ted Kaehler
- Novice Corner by Phil Wyman
- Find Next by Ted Kaehler
- Robin Shank's Utility Scripts
- Scrolling Multiple Fields

If you have a bug, suggestion, comment, or just want to know the best way to do something in HyperCard, you can fill out the form and send it to:

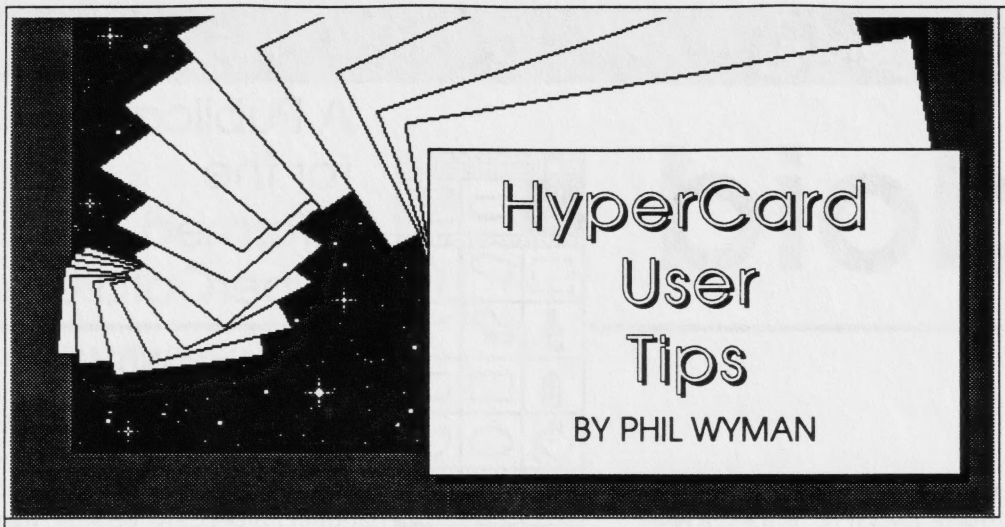
AHUG c/o David Leffler
Apple Computer, Inc.
MS/27-AQ
20525 Mariani Ave.
Cupertino, CA 95014

Or copy the format and Apple-Link™ it to:

HYPERBUG$

EDITOR'S CHOICE

There is a new HyperCard oriented product that I think you should know about. It is called MacRecorder™ from Farallon Computing, Inc. It is a hand-held recording device that plugs directly into the serial port on your Macintosh with no external power supply. One really neat thing about MacRecorder is a stack called HyperSound™. This stack allows you to record a sound and, with a click of a button, install it into any stack complete with its own button. This is really a fantastic and fun tool for painlessly introducing sounds into HyperCard stacks. Additionally, if you have a Macintosh II and two of these devices, you can record in stereo. One last note to Team Hackand-Slash in Hawaii: Thank you for your outstanding contribution to spreading WINDOID around. I love the format and appreciate the way you have taken WINDOID issues and turned them into interesting and fun stacks. Keep up the good work!



HyperCard User Tips

BY PHIL WYMAN

1. Tear off the "Tools" menu. If you double-click on the Eraser tool, you'll erase all card paint. This can be quite shocking if you do it by mistake. If this happens to you, either select UNDO from the Edit menu, press the tilde (~) key, the escape key, or select CMD-Z from the keyboard before doing anything else. This will bring your paint back for you. If CMD-Z doesn't get it back, you can try REVERT from the Paint menu. Revert takes you back to the state the paint was in when you last entered the card. Leaving a card, even for a moment, is the same as saving all the paint elements on that card or choosing Keep from the Paint menu. For fun, double-click on all the other powerful paint tools and see what happens.

2. To initialize a variable in HyperTalk, you should put the assignment of the variable in an "on openStack" handler. You will normally only want to initialize your variables the first time you enter your stack. If this is so, you will need to put the following into your openStack handler...

```
on openStack
  global var
  if var is empty then put 12 into var
end openStack
```

   This way, no matter how many times the user comes into their stack during a HyperCard session, the variable "var" will only be assigned this value the first time, since the second time into the stack the global variable will no longer be empty.

3. If you have a non-SCSI (old HD-20) hard disk or, heaven forbid, a floppy based system, and you have tried to play a long, sophisticated sound in HyperCard, you might notice very poor sound quality because of numerous clicks and pops. These clicks and pops are caused by your disks access time being too slow when HyperCard tries to access them. There is, however, a nice workaround (as suggested by Barbara at Farallon Computing). To stop HyperCard from accessing the disk (doing any processing) while the sound is playing, try this in your script...

```
play "nameOfSound"
wait until the sound is "done"
```

4. A tip from Paul Foraker... If the msg box is already open, and you are typing in a field, there is a way to start typing in the msg box without moving your fingers from the keyboard. Type CMD-M twice and you can now type in the msg box.

5. Tear off the "Tools" windoid. Double-click on "A". This will access different fonts and font attributes for typing Paint Text onto the card or background. If you have already started typing and have not clicked anywhere else, when you change the font you will change the font for everything you just typed. If you have already started typing and you wish to change the font only for what you type next, then click with the mouse at where you want to start before again double-clicking on the "A" in the tool menu.

6. "Not" in HyperTalk works similarly to English. Therefore , you can say...

```
if variable is not "Yes" then go¬
next card
```

   You may also think of "Not" in HyperTalk as a boolean form of the false condition. For example you may have a button which you want to alternately hide and show a field...

```
on mouseup
  set visible of field "fldName" to not¬
  visible of field "fldName"
end mouseup
```

7. If you protect a stack to allow only the browsing userlevel, you can reset it to a different level by pressing on the Command key before clicking on the File menu. You will then be able to choose "Protect Stack" and reset the userlevel.

GESTURE

by Ted Kaehler

Use a gesture to get to a button's script, without having to choose the Button Tool. The gesture consists of pressing and holding the mouse button down inside the button, and moving the cursor in and out of the top edge of the button 2 times! (Rub the top edge of the button.)

Practice until you can do it.

(The trick here is to not interfere with anything else. I don't use the idle task, it can slow things down. The whole thing is in the mouseStillDown message. If your button uses mouseStillDown, you might want to "pass mouseStillDown" at the end of your script.)

Put this into your Home stack script:

```
on mouseStillDown
  --Edits the script of a button if you press and hold the
  --mouse button inside the button, and quickly
  --move in and out of the top edge of the button twice.
  if "button" is not in the name of the target then exit mouseStillDown
  global editTicks, editState
  if (the ticks)-editTicks > 120 then
    put the ticks into editTicks
    put 0 into editState
  else
    get the rect of the target
    if the mouseV < (item 2 of it) then
      if editState = 0 then put 1 into editState
      if editState = 2 then put 3 into editState
    end if
    if the mouseV > (item 2 of it) then
      if editState = 1 then put 2 into editState
      if editState = 3 then put 4 into editState
    end if
  end if
  --put editState
  if editState = 4 then edit script of the name of the target
end mouseStillDown
```

HyperCard Novice Corner

by Phil Wyman

Many novice HyperCard users have asked WINDOID for a beginners' tips section. I thought about this and remembered that most of the questions asked of me at the San Francisco MacWorld Expo started with something like "I just bought a Macintosh and HyperCard came with it. Could you show me what it does?" I have therefore decided to begin this column for WINDOID. If you are a beginning HyperCard user please use the form at the back of WINDOID to send me your questions. I will try to answer some of the most often asked in this column.

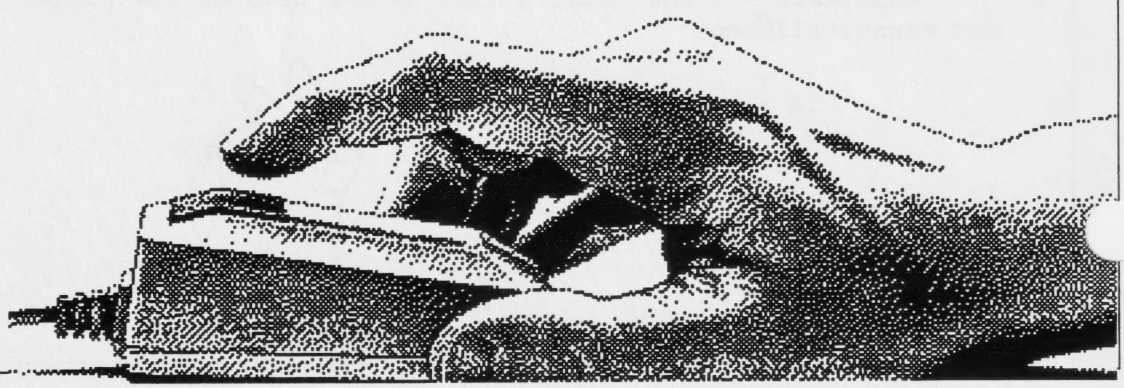
When asked "What is HyperCard?" I would normally begin my answer with something Bill Atkinson said: "HyperCard is a software erector set." In other words, HyperCard is a group of tools you can use to easily create your own Macintosh software applications. The basic tools consist of Buttons, Fields, Graphics, and Cards. Buttons can complete actions when the user clicks on them, Fields can hold text for you, and Graphics can create a mood. A Card is always the exact size of the entire Macintosh Plus screen. When the screen changes, you normally go to another card. When you put a few of these cards together, you have a stack of cards. Stacks are files that you can copy to diskettes and give to your friends or sell to others that need them.

Your button, field, and graphics tools are in your Tools menu. You will see the Tools menu if you are at a high enough "userlevel." To see the userlevel you are currently working with, or to change it, go to the Home card (select Home from the Go menu), and click the left arrow. You will find yourself on the Preferences card. Click on the various Userlevels, Browsing through Scripting, and watch the menubar and Preference card change to reflect your selection.

HyperCard contains a language called HyperTalk. You really don't have to use HyperTalk to create great HyperCard stacks. However, once mastered, HyperTalk will allow you to automatically do almost anything you can do manually in HyperCard. Some beginning HyperTalk users have trouble because they think that you have to type all the HyperTalk commands exactly like they are presented in the on-line Help documentation. This is not so. HyperTalk is very forgiving. For example, the HyperTalk command "doMenu", allows you to select any menu item automatically (this works for Desk Assessories as well.) If you use a menu item that is two words long and/or followed by three periods you may have to put both words and/or the periods between quotes. If you type "doMenu Home" into a button script, the next time you click that button, you will go to the Home Card.

```
on mouseUp
     doMenu Home
end mouseUp

on mouseUp
     doMenu "Open Stack..."
end mouseUp
```



However, you don't have to type "doMenu Home" with the capital and lowercase letters exactly as they appear in the example. We do this in the documentation for easier readability. All the following examples will do exactly the same thing: "domenu home"; "DOMENU HOME"; "doMENU hoME"; or "doMenu Home". Therefore, you should know that in HyperTalk nothing is case sensitive. It never matters whether the words are spelled with an upper or lower case. Even the "Find" command does not care about letter case-sensitivity. If you try to Find "TEST" in HyperCard, you will find all occurrences of the word "test" including "Test"; "test"; "TEST"; and "tesT".

Many beginning HyperCard users have been bothered by the fact that sometimes, when they are typing in a field, the I-beam cursor disappears. They have to repeatedly click in the field to continue typing. This is noticeable in the Home Card or on other Cards where a HyperTalk script updates the time that may appear in another field. When the script updates the time, the cursor is moved into that field and is removed from the field you are working in. To fix this, you either have to use another stack or disable the feature by removing or inactivating the "On idle" HyperTalk script. To inactivate a line of any HyperTalk script you type a double hyphen (- -) before the line.

A hint for moving between fields on a card is to press the TAB key to go from field to field. If you wish to go backwards from field to field, hold the SHIFT key down and press the TAB key.

One really great thing about HyperCard is that it allows non-programmers, the rest of us, to make full use of our computers without having to become programmers. HyperCard allows us to create powerful and meaningful applications with just buttons, fields, cards, and graphics. So don't feel as if you have to learn the HyperTalk language to create your application. Rather, learn to use the basic tools Macintosh and HyperCard gives you to create a stack of cards that can help you in your own area of expertise. Just double-click on "HyperCard" in the finder. You will arrive in the HOME card, which will be your base of operations. Click on the buttons to see what they do. If you get disoriented, you can always choose "Go Home" from the Go menu to go back to your Home card. If the menu isn't showing, hold down the Command key and press the Spacebar to show it again. I hope you have a wonderful time using and enjoying HyperCard.

FIND NEXT

Civilized searching controlled by your script

by Ted Kaehler

The function findNext(theString) goes to the next card containing the string. It uses the find command because its so fast. It counts the number of cards the string has been found on, and lets you do whatever you want on each card it finds. findNext keeps track of the first card it found and tells you when you've gone all the way around the stack.

To use findNext, declare "numberFound" to be a global variable. Set it to zero. findNext("Abe") will return true if it found a new card with "Abe" on it. It returns false if there are no more cards with "Abe". Each time, "numberFound" has the number of cards "Abe" was found on.

Example: Put this into your button script.

```
on mouseUp
-an example of using findNext()
-count the number of cards with a word starting with "Abe" or whatever
  global numberFound
  put 0 into numberFound
  repeat while findNext("Abe") -"Abe" or whatever you choose
  end repeat
  put "Found on" && numberFound && "cards" into message
end mouseUp
```

FIND NEXT (continued)

Put this into your stack script.

```
function findNext key
  -fast searching of a stack under program control.  see example.
  -return true if we found a new occurance, false if done
  -numberFound will have the total we have found
  global numberFound, firstFound
  if (numberFound = 0) or (firstFound is empty)
  then return findFirst(key)

  go next card
  find key

  if the result is "not found" then
    put empty into firstFound
    return false
  end if

  if the ID of this card <> firstFound then
    add 1 to numberFound
    return true   -we found another occurance
  else  -we wrapped around the stack
    put empty into firstFound
    return false
  end if
end findNext

function findFirst key
  -only called by findNext (not called by user)
  -because numberFound = 0, we know this is the first time
  global numberFound, firstFound
  if numberFound <> 0 then
    ask "You must put 0 into numberFound before calling findNext()" with "OK"
    return false
  end if

  find key

  if the result is "not found" then
    put empty into firstFound
    return false
  else   -we found it
    put the ID of this card into firstFound
    put 1 into numberFound
    return true
  end if
end findFirst
```

You can put findNext() and its subroutine, findFirst(), into your home script. If you have stricter conditions on the cards you want to find, do your tests on each card that findNext finds.

HyperCard Utility Scripts

By Robin Shank

The following is a small collection of utility scripts that I find useful in my work.

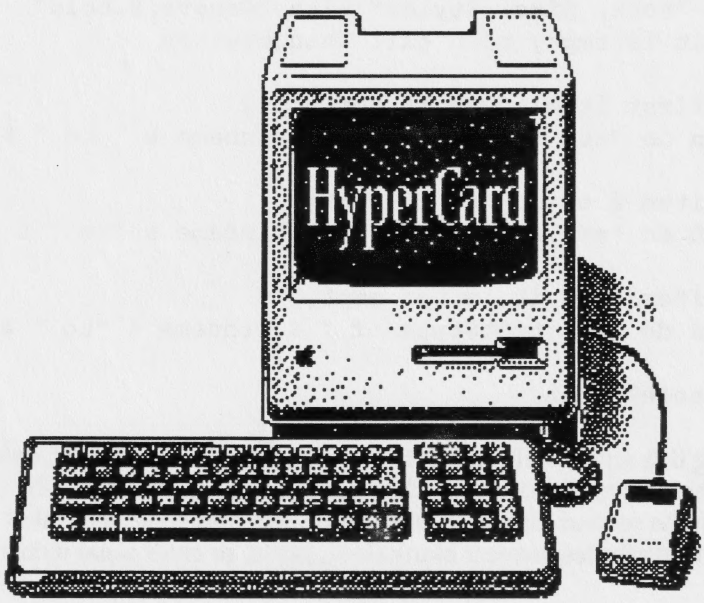
Finding the Finder

When running under MultiFinder, the following script will hide the HyperCard window, the Message window, and any tearoff windoids and put you in the Finder. It is handy when installed into a button on the Home card, especially when you are working on a small-screen Mac. When you choose HyperCard from the Apple menu, the HyperCard windows will reappear exactly as you left them. Note: Hide/Show card window is a new feature of version 1.1, so you must be running that version for this script to work.

```
on mouseUp
  hide card window
  doMenu "finder"
end mouseUp
```

The following is another handy script that uses hide/show card window. This utility allows you to toggle between settings for small and large screen Macs. One useful example is to put the word "setWindow" under "StartUp" in your Home Stack script and put the following script at the end. When you first open HyperCard you can have your windows right where you want them.

```
on setWindow
  answer "Which CPU?" with "Mac II" or "Mac +"
  if it is "Mac II" then
    show menuBar
    show card window at 20,50
    show tool window at 524,198
    show message window at 0,370
  else if it is "Mac +" then
    show card window at 0,0
    hide tool window
    hide pattern window
    show msg at 30,260
    hide msg
  end if
end setWindow
```



Utility Scripts (Continued)

Recovering Lost Buttons

Well, this script won't recover anything you've already lost, but it may save you a headache in the future. Put this script in the Home stack script. When ever you try to get rid of a button this script will complain and give you the chance to get your button back.

```
on deleteButton
  answer "Do you REALLY want to delete that button?" ¬
  with "No, Copy it" or "Yes"
  if it is "No,Copy it" then
    doMenu "copy button"
    doMenu "paste Button"
  end if
end DeleteButton
```

Changing a Button's Font Style

Yes, there are alternatives to those generic round rect buttons with text in good ol' Chicago. Buttons can be assigned a font and font style through a script. If you wish to change the font of a single button, just type set the textFont of card button "fred" to "Geneva" into the message box, and hit return. In addition, you can set the size (set textSize...), style (set textStyle...) alignment (set TextAlign...), and height (set textHeight...), of any button. My current personal favorite is a shadowed button with 10 point Geneva text.

Once you get the hang of changing the text-related attributes of buttons, it can get a bit tedious to set all these styles by typing lines into the msg box. The following script installed into a functionKey and invoked from the message box can be a real finger saver. It will change the button you specify to your choice of font, font size, and font style.

```
on ChangeButton
   ask "Which button?" with "Card button ID " & id of last button
   if it is empty then exit ChangeButton
   put it into btnname

   ask "Font, Size, Style?" with "Geneva,9,bold"
   if it is empty then exit ChangeButton

   if first item of it is not empty
   then do "set textFont of " & btnname & " to " & first item of it

   if item 2 of it is not empty
   then do "set textSize of " & btnname & "to " & item 2 of it

   if item 3 of it is not empty
   then do "set textStyle of " & btnname & "to " & item 3 of it

end changeButton
```

Invoking this script will present two dialogs. In the first dialog, enter the full name of any button on that card. Remember to specify card or background button, and refer to it by name, id or number. (bkgnd btn "New button".) In the second dialog, enter the font, size and style, separated by commas. If you do not wish to change one of the attributes, leave it blank (font, ,style) or enter what it currently is set to.

Utility Scripts (Continued)

An Alternative Solution

Locked fields can behave just like buttons. Create a field, enter your title into it, choose your text style through the Textstyle window, then lock the field. You can now put any "mouse" handler (on mouseUp,etc) into the script of this field and it will behave just like a rect button, with the exception of hiliting.

Which Line Did I Click On?

The following function scripts return which line of a field was clicked on. In this case, "line" is defined as the number of textHeight increments from the top of the field, and not as a line of text ending with a carriage return. Also, the field needs to be locked when you click on it, or you will end up putting an insertion point in the field, not executing a script.

A simple example of how this might be useful:
Put the following handler as well as one of the functions into the script of a field. The field should be locked and contain some lines of text that do not wrap.

```
on mouseUp
  answer "You clicked on line " & quote & line lineclicked()¬
  of card field "newt" & quote
end mouseUp

function lineClicked
  return ((the mouseV - item 2 of the rect of the target) ¬
  div the textHeight of the target) + 1
end lineClicked
```

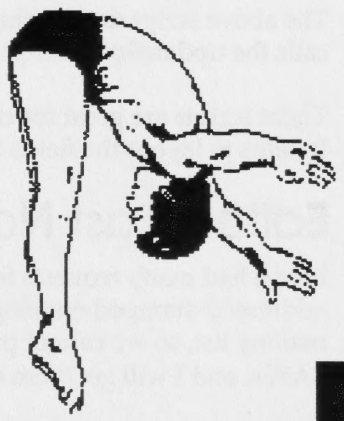
```
If you wish this to work on a scrolling field, it becomes a bit more complex to
account for the possiblity of lines scrolled off the top.

function lineClicked
  return (round((the scroll of the target / the textHeight¬
  of the target) ¬
  +  (((the mouseV - item 2 of the rect of the target) ¬
  div the textHeight of the target) + 1)))
end lineClicked
```

AutoCompaction Will Keep Your Stacks Trim

Install this script into your Home stack script. If any stack you are about to leave needs to be compacted, it will let you know. This is an easy way to ensure you always have lots of room on your hard disk (...for more stacks.)

```
on closeStack
  if the freeSize of this stack >= 15000 then
    answer "This stack has " & round(the freeSize of this¬
       stack/1024) & "k free, Compact?" with  "NO" or "OK"
    if it is "OK" then doMenu "compact stack"
  end if
end closeStack
```



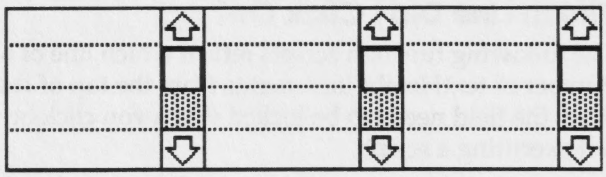
Scrolling Multiple Fields

by Bill Champ

WINDOID is great! It has been a big help to see those example scripts and learn new features and tricks. Especially things that are not shown in Goodman's book (Ed: Danny Goodman - The Complete HyperCard Handbook - Bantam) or other places. Since I've learned so much from WINDOID, I'd like to share a script with the other readers.

How can you get several scrollable fields to all scroll together?

Put these two handlers in your card script.



```
On openCard
  global initScroll
  put 0 into initScroll
  repeat with i = 2 to 5
    set the scroll of card field i to initScroll
  end repeat
end openCard
```

The script above simply sets all the scrollable fields you want (in this case fields 2 thru 5) to zero - elevator at the top of the scroll bar. It also stores that scroll value (zero) in a global variable (initScroll).

```
on updateScroll
  global initScroll
  repeat with i = 2 to 5
    set the scroll of card field i to initScroll
  end repeat
end updateScroll
```

This is the workhorse script. It sets the scroll of the fields you want, all equal to the variable initScroll. initScroll gets changed by the user scrolling one of the fields. That's what the next script does.

Put the following script in each of the scrollable fields.

```
on mouseWithin
  global initScroll
  if the scroll of me <> initScroll then
    get the scroll of me
    put it into initScroll
    updateScroll
  end if
end mouseWithin
```

The above script detects changes in a field's scroll bar and changes the value of the global variable initScroll. Then it calls the updateScroll script (on the card) to change all appropriate fields to the same value.

These scripts are great for creating tabular style fields that can scroll and yet keep all the info lined up between fields. It helps to layout the fields evenly horizontal on the card.

Editor's Last Note:

I have had many requests for back issues of WINDOID. If you would like a copy, or copies, please send a self-addressed-stamped-envelope-(SASE) to the address listed on THE FORM for each issue. AHUG does not have a mailing list, so we cannot put you on ours. If you would like to receive future issues of WINDOID, please send SASEs, and I will get them out to you as time allows. Thank you for your continued interest in WINDOID.

MAKE A DIFFERENCE IN THE WORLD

Tell HyperCard

Date: _____ Phone: _____

Name: _____

MailStop or Address: _____

Versions of:

a. HyperCard _____

b. Associated software _____

_____

c. System _____

d. Finder _____

e. ImageWriter _____

f. LaserWriter _____

g. Any others _____

_____

Type of Macintosh: _____

Peripherals: _____

SEND TO:

AHUG
c/o David Leffler
Apple Computer, Inc.
MS/27-AQ
20525 Mariani Ave.
Cupertino, CA 95014

or
copy the format and
Apple-Link™ it to
HYPERBUG$

Description of problem, suggestions, or comments:

_____

_____

_____

_____

_____

_____

please write on the back if necessary